METHODS AND SYSTEMS FOR MANAGEMENT AND CONTROL OF AN AUTOMATION CONTROL MODULE

BACKGROUND OF THE INVENTION

[0001] This invention relates generally to automation control modules (ACMs) and more particularly to management and control of ACMs.

[0002] Known ACM systems access ACM data using an input/output (I/O) module connected to a backplane on the ACM. The I/O module increases the cost of the system and uses additional space in the ACM system cabinet. In addition, if the backplane includes a plurality of modules, the I/O module may respond more slowly.

[0003] Known web-enabled ACM systems also provide pre-defined web pages that contain ACM data in a format determined by the manufacturer. Pre-defined web pages are inflexible and may be intolerable to many users. Furthermore, known ACM systems that include a web server use the ACM's central processing unit (CPU) to run the web server and the transmission control protocol (TCP)/internet protocol (IP) stack, thereby degrading performance of the CPU.

BRIEF DESCRIPTION OF THE INVENTION

[0004] In one aspect, a web-enabled automation control module (ACM) is provided that includes an ACM central processing unit (CPU) and a web and file transfer system electrically connected to the ACM CPU, and embedded within the ACM. The system is configured to process hypertext transfer protocol (HTTP) requests from a network.

[0005] In another aspect, an automation control module (ACM) system is provided that includes an ACM, a network, a web-enabled computer electrically connected to the network, and a web and file transfer subsystem

electrically connected to the ACM and the network. The subsystem is configured to store at least one user-defined web page file.

[0006] In yet another aspect, a method is provided for management and control of an automation control module (ACM). The ACM includes an ACM central processing unit (CPU) and a web and file transfer system embedded within the ACM. The web and file transfer system is electrically connected to a network. The method includes electrically connecting the web and file transfer system to the ACM CPU, and processing hypertext transfer protocol (HTTP) requests from the network using the web and file transfer subsystem.

[0007] In a further aspect, a method is provided for management and control of an automation control module (ACM) using an ACM system. The ACM system includes an ACM, a network, a web-enabled computer electrically connected to the ACM, and a web and file transfer subsystem. The method includes electrically connecting the web and file transfer subsystem to the ACM and the network, and storing at least one user-defined web page.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Figure 1 is a block diagram illustrating one embodiment of an ACM system of the present invention.

[0009] Figure 2 is a flow chart illustrating one embodiment of a method for management and control of an automation control module using an ACM system.

[0010] Figure 3 is a flow chart illustrating another embodiment of a method for management and control of an automation control module using an ACM system.

[0011] Figure 4 is a flow chart illustrating another embodiment of a method for management and control of an automation control module using an ACM system.

[0012] Figure 5 is an example of an ACM HTML tag comment.

[0013] Figure 6 is an alternative example of an ACM HTML tag comment.

[0014] Figure 7 is an example of an HTML web page for retrieving a mode of the ACM of the present invention using HTML and ACM tag functions.

[0015] Figure 8 is the HTML for the web page illustrated in Figure 6.

[0016] Figure 9 is an HTML for a web page that displays the ACM mode requested in the HTML illustrated in Figure 7.

[0017] Figure 10 is an example of an HTML for a web page for retrieving a mode of the ACM of the present invention using HTML, Javascript, and ACM tag functions.

DETAILED DESCRIPTION OF THE INVENTION

[0018] Set forth below are descriptions of embodiments of methods and systems for control and management of an automation control module (ACM). The methods and systems facilitate viewing and controlling ACM data through standard networks, protocols, and browsers, developing and downloading user-defined web pages that include ACM data, and controlling the access level to the ACM and user-defined web pages.

[0019] The methods and systems are not limited to the specific embodiments described herein. In addition, components of each system and steps of each method can be practiced independent and separate from other components and steps described herein. Each component and step can also be used in combination with other components and steps.

[0020] As used herein, the term ACM refers to any device used to control the automation of an activity, including but not limited to PLCs, computer numeric controls (CNCs), motion control products, home automation products, and

commercial automation products, for example controls for automated teller machines or car wash systems. As used herein, ACM data includes different types of data within an ACM system 10 that control operation of ACM system 10. ACM data includes, but is not limited to, user logic programs, user program memory, ACM status and statistics, ACM faults, setting ACM operating states, setting privilege levels, and any other useful ACM information.

[0021] Figure 1 illustrates, in block diagram form, hardware architectures that can be utilized in conjunction with an ACM management and control system. The system can be implemented on many different platforms and utilize many different architectures. The architectures illustrated in Figure 1 are exemplary only.

[0022] Figure 1 is a block diagram illustrating one embodiment of ACM system 10. System 10 includes an eWeb ACM 12, a web and file transfer subsystem 14, and a web-enabled computer 16. EWeb ACM 12 includes an ACM CPU 18 that carries out ACM functions, for example user logic and function block executions, input/output (I/O) scanning, and communications to other devices. ACM CPU 18 includes a CPU system memory 20 electrically connected to CPU 18 and, in one embodiment, contains both the operating system (not shown) for ACM CPU 18 and a user's program and data. In one embodiment, an ACM I/O backplane interface 22 is connected to ACM CPU 18, and provides an interface between ACM CPU 18 and an ACM backplane 24 connected to interface 22. ACM backplane 24 provides a physical and electrical means for connecting various I/O or other input modules 26, for example communications or motion modules, into eWeb ACM 12. ACM backplane 24 facilitates the exchange of data between modules 26 and ACM CPU 18. In one embodiment, one or more modules 26 provide an interface for real world inputs (not shown), such as limit or proximity switch status, position of an object, temperature, or pressure, to ACM CPU 18 as parameters for logic or function block execution. In another embodiment, one or more modules 26 provide an interface to real world outputs (not shown) as commanded by ACM CPU 18 to control output devices (not shown), such as actuators, contactors, or solenoids.

[0023] Web-enabled computer 16 is electrically connected to a network 28. Network 28 includes the physical medium and intermediate devices (not shown), such as routers, and switches, that connect computer 16 to eWeb ACM 12. In one embodiment, network 28 is a wide area network (WAN), such as the Internet. In an alternative embodiment, network 28 is a local area network (LAN), such as an Intranet. A user 30 accesses, such as dialing into, or directly logging onto, an Intranet or the Internet to gain access to eWeb ACM 12. In one embodiment, computer 16 includes a web browser, and eWeb ACM 12 is accessible to computer 16 via the Internet. Computer 16 is interconnected to the Internet through many interfaces including a different network (not shown), such as a WAN or a LAN, dial in connections, cable modems and special high-speed ISDN lines. Computer 16 is any device capable of interconnecting to the Internet, including a web-based telephone or other web-based connectable equipment.

[0024] Computer 16 displays PLC data on at least one web page (not shown), and retrieves web page files (not shown) stored on a web page file database 32 embedded within web and file transfer subsystem 14. Web page files are text files that may contain hypertext markup language (HTML), Javascript, and/or references to other files, such as image files to be displayed with the web page or Java Applets. In another embodiment, web page files include ACM tag functions that reference ACM data stored in CPU system memory 20. The tag facilitates the exchange of data between ACM CPU 18 and a web server 34 embedded within web and file transfer subsystem 14. Further, the tag provides a generic mechanism for user 30 to display and/or control ACM data with a standard browser. In one embodiment, computer 16 includes web authoring tools and/or text editors that, along with user input, are utilized to create and modify web page files.

[0025] Users 30 include at least one person who views and/or controls ACM data from computer 16. In one embodiment, users 30 include a person who created a web page file. Web and file transfer subsystem 14 is electrically connected to ACM CPU 18, CPU system memory 20, and network 28. Subsystem 14 is shown in Figure 1 to be embedded within eWeb ACM 12. In an alternative

embodiment, subsystem 14 is contained in a separate module connected to backplane 24. Web and file transfer subsystem 14 includes web page file database 32, web server 34, a file transfer server 36, and a network interface 38 that provides the lower level protocols (TCP/IP) and physical hardware connections to network 28. File transfer server 36 is electrically connected to web page file database 32 and network interface 38, and transfers web page files and associated elements between web page file database 32 and computer 16. File transfer server 36 facilitates downloading customizable user 30 defined web pages to eWeb ACM 12 as described below. In one embodiment, file transfer server 36 is a file transfer protocol server.

[0026] Web server 34 is electrically connected to web page file database 32, network interface 38, and ACM CPU 18. Web server 34 receives and processes hypertext transfer protocol (HTTP) requests to send web pages to computer 16 and, based upon the requests, sends the requested web page to computer 16. If the requested web page includes a tag function, web server 34 parses and executes the tag function and either embeds ACM data within a web page file thereby displaying the web page on a browser on computer 16, or transmits ACM data to ACM CPU 18. In one embodiment, web server 34 transfers ACM data to ACM CPU 18 to control operation of eWeb ACM 12.

[0027] In one embodiment, user 30 must enter a valid user name and valid user password to access eWeb ACM 12 and web and file transfer subsystem 14. The user name and user password correspond to a user profile stored in web page file database 32. User 30 configures the number of web and file transfer TCP connections (not shown) using computer 16. A value of zero allows user 30 to disable the web and file transfer TCP connections.

[0028] Figure 2 is a flow chart 50 illustrating a method 52 for management and control of eWeb ACM 12 (shown in Figure 1). Method 52 includes electrically connecting 54 web server 34 (shown in Figure 1) to ACM CPU 18 (shown in Figure 1). Web server 34 receives 56 HTTP requests from network 28 (shown in Figure 1) and processes 58 the HTTP requests. In one embodiment, web server 34 processes 58 the HTTP requests, reads 60 at least one web page file (not shown),

parses 62 the web page file for tag functions, uses form data from the HTTP request to transfer 64 ACM data to ACM CPU 18 to control operation of eWeb ACM 12, embeds 66 the status of the tag function within the at least one web page file, and sends 68 the at least one web page file through network 28. In another embodiment, web server 34 processes 58 the HTTP requests, reads 60 at least one web page file from database 32, requests 70 ACM data from ACM CPU 18 based on parsing the web page file for tag functions and applying form data from the HTTP request, and receives 72 ACM data from ACM CPU 18. Further, web server 34 embeds 74 the received ACM data within the at least one web page file and sends 68 the at least one web page file through network 28.

[0029] Figure 3 is a flow chart 80 illustrating a method 82 for management and control of eWeb ACM 12 (shown in Figure 1) using ACM system 10 (shown in Figure 1). Method 82 includes requiring 84 user 30 to input a valid user name and valid user password to access web and file transfer subsystem 14 (shown in Figure 1) and allowing 86 user 30 to transfer user defined or other web page files to database 32. In another embodiment, method 82 includes allowing 88 user 30 to configure the number of web and file transfer TCP connections (not shown) using an ACM configuration package (not shown). The web and file transfer TCP connections are disabled 90 when user 30 configures zero of the web and file transfer TCP connections.

Figure 4 is a flow chart 100 illustrating a method 102 for management and control of eWeb ACM 12 (shown in Figure 1) using ACM system 10 (shown in Figure 1). Method 102 includes electrically connecting 104 web and file transfer subsystem 14 (shown in Figure 1) to eWeb ACM 12, allowing 106 user 30 to create a user-defined web page file (not shown) using computer 16 (shown in Figure 1) and file transfer server 36 (shown in Figure 1), and storing 108 at least one user-defined web page file in database 32 (shown in Figure 1). File transfer server 36 reads 110 the at least one user-defined web page file stored in database 32, transfers 112 the at least one user-defined web page file to computer 16, and displays 114 the at least one user-defined web page file on computer 16. Computer 16 and file transfer server 36 allow

116 user 30 to modify the at least one user-defined web page file, and file transfer server 36 writes 118 to the at least one user-defined web page file.

[0030] Examples

[0031] Figure 5 is an example of an ACM HTML tag comment syntax 200 that includes a specified defined function 202. Tag comment 200 also includes a plurality of attributes 204, which may be function-specific parameters or general attributes, such as data display requirements.

[0032] Figure 6 is an alternative example of an ACM HTML tag comment 210 that includes a specified defined function 212. Tag comment 210 also includes a plurality of attributes 214.

[0033] Figure 7 is an example of a web page 220 for retrieving a mode of eWeb ACM 12 using HTML and ACM tag functions.

[0034] Figure 8 is the HTML 230 for web page 220. HTML 230 includes an ACM tag function 232 that causes web server 34 to retrieve the ACM mode from eWeb ACM 12.

[0035] Figure 9 is an example of an HTML 240 for a web page that displays the ACM mode requested in HTML 230. HTML 240 includes a return string value 242 from tag function 232. HTML 240 does not include the ACM tag information.

[0036] Figure 10 is an example of an HTML 250 for a web page for retrieving a mode of eWeb ACM 12 using HTML, Javascript, and ACM tag functions. HTML 250 includes an ACM tag function 252 that causes web server 34 to retrieve the mode of eWeb ACM 12.

[0037] ACM system 10 provides reduced system hardware costs, rapid development of custom ACM monitoring and control tools that reduce implementation costs, and fast response time accessing ACM data with low impact on other critical real-time ACM functions such as ACM sweep time, thereby reducing

production costs. In addition, ACM system 10 facilitates rapid access to ACM data on standard devices such as a web browser on computer 16 or PDA via a standard network.

[0038] While the invention has been described in terms of various specific embodiments, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the claims.